**Практическая работа № 1**

**Функции централизованного диспетчера прерываний на примере Windows NT**

1. **Цель работы**
   1. Изучить функции централизованного диспетчера прерываний.
   2. Научиться определять какие прерывания использует то, или иное устройство ПЭВМ.
2. **Средства обучения**
   1. Методические указания по выполнению лабораторной работы.
   2. Курс лекций.
3. **Ход работы**

Дайте ответы на каждое задание

* 1. **Дайте определение ядра операционной системы.**

Ядро операционной системы (ОС) — это центральный компонент, который управляет всеми ресурсами системы, включая процессоры, память и устройства ввода-вывода. Оно обеспечивает интерфейс между аппаратным обеспечением и пользовательскими приложениями, выполняя такие функции, как управление процессами, памятью и обработка прерываний

* 1. **Каким образом обеспечивается высокая скорость работы ОС.**

Высокая скорость работы операционной системы достигается за счет:

* Оптимизации алгоритмов: Эффективные алгоритмы планирования процессов и управления памятью минимизируют время ожидания.
* Использования кэширования: Кэширование данных и инструкций позволяет ускорить доступ к часто используемым ресурсам.
* Мультипрограммирования: Многозадачность и параллельное выполнение процессов позволяют более эффективно использовать ресурсы процессора
  1. **Перечислите на какие группы обычно подразделяются вспомогательные модули ОС.**

Вспомогательные модули операционной системы обычно подразделяются на следующие группы:

* Драйверы устройств: обеспечивают взаимодействие с аппаратным обеспечением.
* Системные библиотеки: предоставляют интерфейсы для программного обеспечения.
* Службы: выполняют фоновые задачи, такие как управление сетью или безопасностью
  1. **Поясните, что такое привилегированный режим.**

Привилегированный режим — это режим работы процессора, в котором выполняются операции с высоким уровнем доступа к системным ресурсам. В этом режиме ядро ОС может выполнять критически важные операции, такие как управление памятью и обработка прерываний, что недоступно для пользовательских приложений

* 1. **Поясните основную идею микроядерной структуры ОС.**

Основная идея микроядерной структуры заключается в минимизации функций, выполняемых ядром, путем переноса большинства служб в пользовательское пространство. Это повышает устойчивость системы к сбоям и упрощает обновление компонентов, однако может снижать производительность из-за необходимости переключения между режимами

* 1. **Поясните назначение серверов операционной системы.**

Сервера операционной системы выполняют специализированные функции, такие как управление файлами, сетевыми соединениями и устройствами. Они обеспечивают взаимодействие между различными компонентами ОС и приложениями, предоставляя необходимые сервисы по запросу

* 1. **Перечислите основные недостатки и достоинства микроядерной структуры ОС.**

Достоинства:

* Повышенная надежность: сбой одного компонента не приводит к сбою всей системы.
* Гибкость: легко добавлять или изменять модули без перезагрузки системы.

Недостатки:

* Снижение производительности: частые переключения между режимами могут замедлить выполнение задач.
* Сложность разработки: требует более сложной архитектуры для взаимодействия между компонентами.
  1. **Перечислите основные слои ядра ОС.**

Основные слои ядра операционной системы включают:

* Управление процессами: Создание и завершение процессов.
* Управление памятью: Выделение и освобождение памяти.
* Обработка прерываний: Реакция на события от устройств.
* Управление устройствами: Взаимодействие с аппаратным обеспечением
  1. **Поясните от чего зависит переносимость ОС.**

Переносимость операционной системы зависит от:

* Абстракции аппаратного обеспечения: Наличие универсальных интерфейсов для различных платформ.
* Используемых языков программирования: Языки с высокой переносимостью (например, C) облегчают адаптацию под разные архитектуры.
* Стандартов API: Соответствие стандартам позволяет разработчикам легко переносить приложения между системами
  1. **Перечислите основные классы прерываний.**

Основные классы прерываний включают:

* Аппаратные прерывания: генерируются устройствами (например, клавиатура).
* Программные прерывания: инициируются программами (например, системные вызовы).
* Таймерные прерывания: используются для управления временем выполнения процессов
  1. **Перечислите основные способы прерываний.**

Способы обработки прерываний включают:

* Маскирование прерываний: Отключение определенных прерываний для предотвращения конфликтов.
* Приоритетное обслуживание: Обработка более важных прерываний в первую очередь.
* Обработка в фоновом режиме: Обработка менее критичных задач после завершения текущих операций
  1. **Поясните механизм приоритезации прерываний.**

Механизм приоритезации прерываний позволяет системе обрабатывать более важные прерывания раньше менее важных. Это достигается путем назначения каждому прерыванию уровня приоритета; при возникновении нескольких прерываний система обрабатывает их в порядке убывания приоритета

* 1. **В чем заключается маскирование прерываний.**

Маскирование прерываний — это процесс отключения определенных прерываний для предотвращения их обработки во время выполнения критических задач. Это позволяет избежать конфликтов и обеспечивает стабильность работы системы

* 1. **Перечислите основные задачи, выполняемые диспетчером прерываний.**
* Обработка входящих прерываний.
* Определение приоритетов для обработки.
* Передача управления соответствующим обработчикам
  1. **Перечислите основные уровни запросов прерываний.**
* Высокий уровень: обрабатывает критические задачи (например, аппаратные сбои).
* Низкий уровень: обрабатывает менее важные задачи (например, ввод данных от пользователя).
* Диспетчерский/DPC уровень: обрабатывает отложенные задачи после завершения текущих операций
  1. **Поясните назначение уровня запроса «Диспетчерский/DPC».**

Уровень запроса «Диспетчерский/DPC» используется для обработки задач с низким приоритетом после завершения более критичных операций. Это помогает оптимизировать использование ресурсов и улучшить отклик системы

* 1. **Поясните понятие «Системный вызов».**

Системный вызов — это механизм, позволяющий приложениям запрашивать услуги у ядра операционной системы. Он служит интерфейсом между пользовательским пространством и пространством ядра, обеспечивая доступ к системным ресурсам

* 1. **Поясните идею децентрализованной обработки системных вызовов.**

Идея децентрализованной обработки системных вызовов заключается в распределении обработки вызовов между несколькими компонентами или серверами. Это позволяет повысить эффективность и устойчивость системы за счет уменьшения нагрузки на ядро

* 1. **Поясните идею централизованной обработки системных вызовов.**

Централизованная обработка системных вызовов подразумевает выполнение всех запросов через одно ядро или сервер. Это упрощает архитектуру системы, но может привести к узким местам в производительности из-за высокой нагрузки на центральный компонент

* 1. **В чем заключается отличие синхронного и асинхронного режимов обработки системных вызовов.**

Синхронный режим обработки системных вызовов требует завершения операции перед продолжением выполнения программы, тогда как асинхронный режим позволяет программе продолжать выполнение независимо от завершения операции. Это повышает общую производительность и отзывчивость приложений

1. **Вывод по работе:** я узнал о различных аспектах операционных систем, включая структуру ядра, механизмы управления памятью и процессами, а также особенности обработки прерываний и системных вызовов; Я понял, как функционируют основные компоненты операционной системы и как они взаимодействуют друг с другом. Осознал важность привилегированного режима для обеспечения безопасности и стабильности системы, а также роль вспомогательных модулей в расширении функциональности ОС; Я осознал, что глубокое понимание принципов работы операционных систем позволяет не только эффективнее использовать их возможности, но и разрабатывать более надежные и производительные приложения.